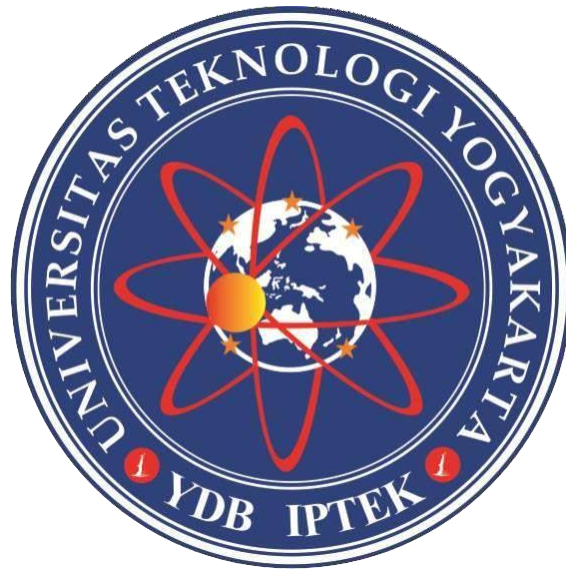


RESPONSI SOP-V

Dosen Pengampu : Iwan Hartadi Tri Untoro, S.T., M.Kom.

Asisten Dosen : Galang Aidil Akbar



Disusun Oleh:

Aulia Nugraheni (5200411114)

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2021**

1. Simulasi Manajemen RAM

Source Code :

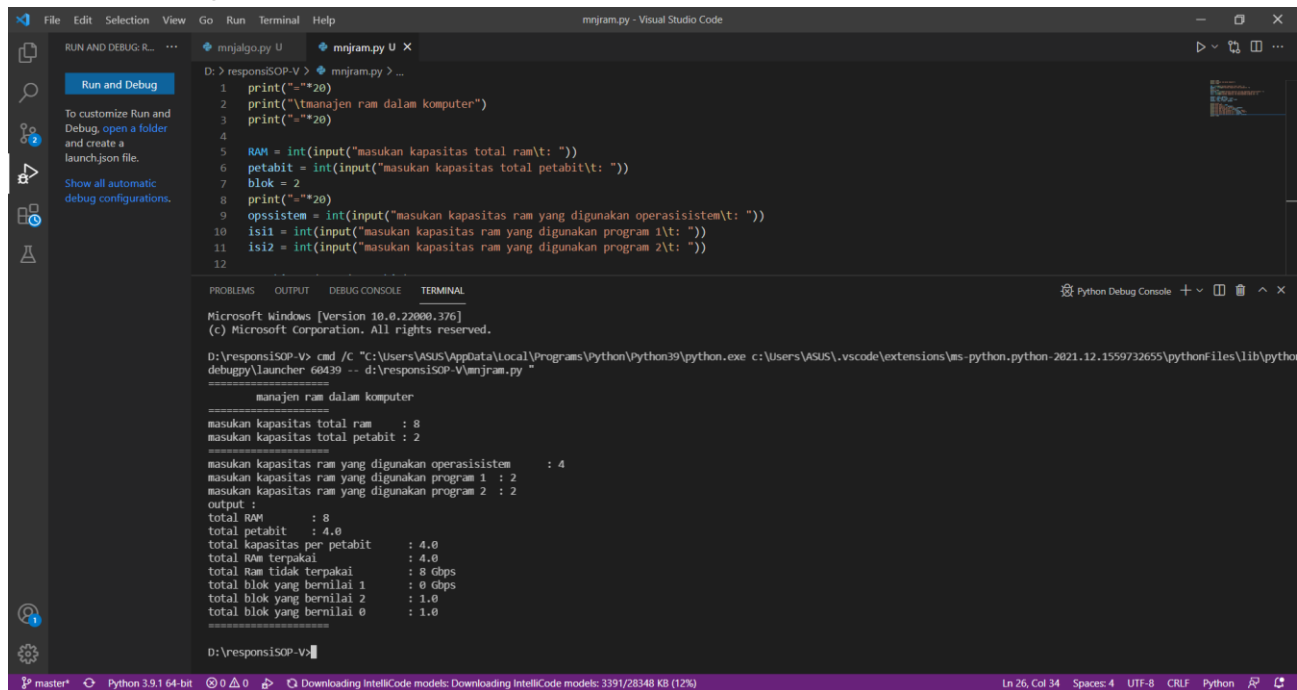
```
print("="*20)
print("\tmanajemen ram dalam komputer")
print("="*20)

RAM = int(input("masukan kapasitas total ram\t: "))
petabit = int(input("masukan kapasitas total petabit\t: "))
blok = 2
print("="*20)
opssistem = int(input("masukan kapasitas ram yang digunakan operasisistem\t: "))
isi1 = int(input("masukan kapasitas ram yang digunakan program 1\t: "))
isi2 = int(input("masukan kapasitas ram yang digunakan program 2\t: "))

petabit = (RAM / petabit)
proter = (opssistem + isi1 + isi2)
prot dk = (RAM - opssistem - isi1 - isi2)
jmlblok = (isi1 + isi2) / petabit

print("output :")
print("total RAM\t :",RAM)
print("total petabit\t :",petabit)
print("total kapasitas per petabit\t :",petabit)
print("total RAM terpakai\t\t :",petabit)
print("total Ram tidak terpakai\t :",proter, "Gbps")
print("total blok yang bernilai 1\t :",prot dk, "Gbps")
print("total blok yang bernilai 2\t :",jmlblok)
print("total blok yang bernilai 0\t :",blok - jmlblok)
print("="*20)
```

Hasil Running :



```
D:\> responsiSOP-V > mnjram.py > ...
1  print("="*20)
2  print("\tmanajemen ram dalam komputer")
3  print("="*20)
4
5  RAM = int(input("masukan kapasitas total ram\t: "))
6  petabit = int(input("masukan kapasitas total petabit\t: "))
7  blok = 2
8  print("="*20)
9  opssistem = int(input("masukan kapasitas ram yang digunakan operasisistem\t: "))
10 isi1 = int(input("masukan kapasitas ram yang digunakan program 1\t: "))
11 isi2 = int(input("masukan kapasitas ram yang digunakan program 2\t: "))
12

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

D:\responsiSOP-V> cd /C "C:\Users\ASUS\AppData\Local\Programs\Python\Python39\python.exe -c \Users\ASUS\.vscode\extensions\ms-python.python-2021.12.1559732655\pythonFiles\lib\python
debugpy\launcher 66439 -- d:\responsiSOP-V\mnjram.py"
=====
manajemen ram dalam komputer
=====
masukan kapasitas total ram      : 8
masukan kapasitas total petabit : 2
=====
masukan kapasitas ram yang digunakan operasisistem      : 4
masukan kapasitas ram yang digunakan program 1 : 2
masukan kapasitas ram yang digunakan program 2 : 2
output :
total RAM      : 8
total petabit  : 4.0
total kapasitas per petabit      : 4.0
total RAM terpakai      : 4.0
total RAM tidak terpakai      : 8 Gbps
total blok yang bernilai 1      : 0 Gbps
total blok yang bernilai 2      : 1.0
total blok yang bernilai 0      : 1.0
=====
D:\responsiSOP-V>
```

2. Simulasi manajemen penjadwalan dengan algoritma Round Robin

Source Code :

```
import os
listprogram = []

def fungsi(waktu_selesai,jatah,programlist):
    start = 0
    while start < waktu_selesai:
        for i,data in enumerate(programlist):
            proses = data[0]
            waktu_proses = data[1]
            waktu_sisa = waktu_proses - jatah

            if(waktu_proses >= jatah):
                print(proses,'-> Detik Ke -',start, ' s.d. - Detik Ke -', start +
jatah )
            else:
                print(proses,'-> Detik Ke -',start, ' s.d. - Detik Ke -', start +
waktu_proses )

            if(waktu_proses >= jatah):
                start += jatah
            else:
                start += waktu_proses

        if( waktu_sisa > 0):
```

```

        listprogram.append([proses,waktu_sisa])

os.system("cls")
total_proses = int(input('Total Proses : '))

for i in range(total_proses):
    proses = input('Nama Proses : ')
    print("")
    waktu = int(input('Waktu Proses : '))
    print("")
    data_list = [proses,waktu]
    listprogram.append(data_list)

jatah = int(input('Jatah Waktu : '))

time_complete = 0
for i in listprogram:
    time_complete += i[1]

print("")
fungsi(time_complete,jatah,listprogram)

```

Hasil Running :

The screenshot displays the Visual Studio Code interface with a Python file named 'mnjalgo.py'. The code defines a function 'fungsi' that takes 'waktu_selesai', 'jatah', and 'programlist' as arguments. It calculates the remaining time for each process by subtracting the share of time from the total time. The terminal output shows the user inputting 3 for total processes, and then names and times for three processes: A1 (10), A2 (5), and A3 (7). The share of time is input as 15. The final output shows the remaining time for each process: A1 has 0 left, A2 has 10 left, and A3 has 15 left.

```

1  import os
2  listprogram = []
3
4  def fungsi(waktu_selesai,jatah,programlist):
5      start = 0
6      while start < waktu_selesai:
7          for i,data in enumerate(programlist):
8              proses = data[0]
9              waktu_proses = data[1]
10             waktu_sisa = waktu_proses - jatah
11
12

```

Terminal Output:

```

D:\responsiSOP-V> mnjalgo.py > ...
Total Proses : 3
Nama Proses : A1
Waktu Proses : 10
Nama Proses : A2
Waktu Proses : 5
Nama Proses : A3
Waktu Proses : 7
Jatah Waktu : 15
A1 -> Detik Ke - 0 s.d. - Detik Ke - 10
A2 -> Detik Ke - 10 s.d. - Detik Ke - 15
A3 -> Detik Ke - 15 s.d. - Detik Ke - 22
D:\responsiSOP-V>

```